AJAX-Tags

Descripción

La librería AjaxTags es un conjunto de tags JSP que nos simplifican el uso de JavaScript asíncrono y XML (AJAX) en páginas con servidores Java.

Esto hace posible que los desarrolladores Java no necesiten entrar en profundidad en el código de los Javascript para implementar un formulario web tipo AJAX, sino que pueden usar los modelos definidos en la librería AjaxTag, facilitando por tanto su labor.

AjaxTags es diferente de la mayoría de librerías de Ajax, ya que proporciona un acercamiento declarativo a Ajax. Configuras las acciones de Ajax que quieres que sucedan cuando suceda el evento que hayas configurado.

Lo único que tenemos que hacer para utilizar AjaxTags es definir el acontecimiento Ajaz que queremos que se produzca cuando ejecutemos un evento sobre un elemento de la página. Podemos asociar eventos sobre cualquier elemento de la página.

Esta librería nos provee de varias herramientas como pueden ser autocompletar, carga de combos en función de otras, inserción de html en sólo una parte de la página, etc.

Esta ficha presenta el contenido acerca de AJAX-Tags, se muestran a continuación enlaces directos a distintos aspectos:

- Características
- Ventajas e Inconvenientes
- Versiones Recomendadas
- Prerequisitos e Incompatibilidades
- Modo de Empleo
- Enlaces de Interés
- Ejemplos de uso

Características

AjaxTags se encarga tanto del Javascript necesario en la parte del cliente como de los mecanismos de respuesta en el servidor. Dentro de la librería AjaxTags podemos encontrar entre muchas otras las siguientes funciones:

- Auto-complete: recupera una lista de valores que coinciden con la cadena de caracteres introducidos en un formulario de texto a la vez que el usuario los va escribiendo.
- Callout: muestra un área o ventana emergente al hacer clic sobre un elemento de la página sensible al contexto al tener asociado algún tipo de evento de ratón.
- Select/DropDown: al usar esta característica, el sistema trabaja con dos combos, de tal forma que al realizar una selección en el primer menú desplegable, se cargan resultados en el segundo menú en función de la primera selección. Por ejemplo: modelos de coches en función de una marca determinada.
- Form/Update: se actualiza el valor de un campo de texto vacío calculándose a partir de los valores de otros campos del formulario.
- Toggle:intercambia el valor de variables ocultas de un formulario y al mismo tiempo visualiza una imagen en lugar de otra.

- Editor: crea un editor en la misma ventana.
- Tab Panel / Tab: activa una serie de propiedades de ventana basadas en AJAX.

Terminología básica

En esta sección vamos a describir los elementos comunes para cualquier definición de AjaxTags:

baseUrl

Url que queremos que ejecute la librería AjaxTags cuando se produzca el evento.

eventType

Tipo de evento que queremos que ejecute la petición al servidor.

source

Identificador del elemento html que queremos que sea el encargado de lanzar el evento, es decir, es el elemento sobre el que AjaxTags escuchará el evento definido en la propiedad eventType.

sourceClass

La propiedad sourceClass será la encargada de almacenar el nombre una clase css. Sobre los elementos que utilicen esta clase, serán los posibles disparadores del evento ajax que queremos lanzar. Es utilizado para asociar un evento ajax a más de un lanzador.

target

Con esta propiedad se identificará el elemento html destino de la acción de AjaxTags, es decir, será el contenedor de la información solicitada al servidor.

parameters

En esta propiedad se guardarán los parámetros que serán enviados en la request para que sean accedidos desde el servlet encargado de realizar la consulta solicitada. Estos parámetros estarán formados por pares atributo-valor separados por coma.

preFunction

En esta propiedad se indica el nombre de la función javascript que deseamos sea ejecutada antes de realizar la petición al servidor. Normalmente se utiliza para realizar alguna preparación antes de la petición.

postFunction

Es la propiedad que definirá el nombre de la función javascript que será ejecutada cuando el servidor devuelva el resultado de la consulta solicitada. Puede usarse para encadenar con otros eventos.

errorFunction

Es la propiedad que almacena la función javascript que se ejecutará en caso de que se produzca una excepción en el servidor.

Ventajas e Incovenientes

En la ficha general de AJAX hay una comparativa de los distintos framework AJAX analizados. Por este motivo, a continuación describimos las ventajas del uso de AjaxTags:

- No hay que programar Javascript, a menos que necesites realizar alguna prefunction o postfunction.
- Es muy fácil agregar las funciones de Ajax a una página existente sin cambiar ningún código existente y sin agregar mucho más que algunas etiquetas.
- Está bien documentado con un ejemplo disponible.

Algunos puntos débiles de AjaxTags son los siguientes:

- AjaxTags no te cuenta absolutamente nada sobre algún problema que suceda en el servidor, esto lo tienes que implementar tu.

Procedemos a realizar una comparativa entre AJAX y su principal competidor, Flash.

Entre las principales ventajas de AJAX se pueden enumerar las siguientes:

- Basado en HTML: AJAX se basa en XHTML, por lo que es fácilmente estandarizable.
- Accesible: la integración de lectores de pantalla, las posibilidades de modificar el tamaño del texto por el usuario, etc.
- Integración con CSS: es bastante sencillo separar datos de presentación por lo que las aplicaciones son más escalables.
- Ligereza: al ser una página con sólo HTML es más ligera que un front-end con Flash.
- Motores de búsqueda: al basar los interfaces en HTML, son fácilmente indexados por los motores de búsqueda.

Como principal desventaja tenemos que el usuario puede haber desactivado Javascript en el navegador, así como que se depende de un objeto propietario para las comunicaciones asíncronas.

Respecto a Flash, sus principales ventajas son su presencia prácticamente universal en toda la web, así como su excelente calidad visual, pero por contra se depende de un plug-in propietario, así como de una menor accesibilidad y algo realmente molesto, como son los banners y el botón de ##skip intro## inherente a cualquier animación flash.

Versiones Recomendadas

La última versión liberada es la 1.3, algunas de sus principales mejoras respecto a otras versiones son las siguientes:

- Solución de diversos problemas de la función autocompletar con Internet Explorer.
- Se añaden las funciones anchor, area y displayTag.

Prerequisitos e Incompatibilidades

Prerequisitos

Procedemos a describir algunos prerequisitos detectados para la instalación en nuestro sistema de AJAXTags:

- La versión del JDK debe ser la 1.5 o posterior.
- El Servlet container running debe ser la versión 2.3 o posterior y la versión de JSP la 1.0 o posterior.
- En las páginas donde se utilice la funcionalidad de AjaxTags, serán necesarias también las siguientes librerias javascript con sus consiguientes versiones:
  - Prototype versión 1.5.0.
  - Scriptaculous versión 1.7.0.
  - OverLIBMWS.

Incompatibilidades

El código Java es independiente del sistema operativo y no está vinculado a ningún componente determinado, está implementado usando el JDK 1.4 y requiere un contenedor de Servlets para funcionar. El código Javascript se ejecuta sin problemas al menos en las versiones 1.0 de Firefox y 5.0 de Internet Explorer. Otras incompatibilidades no se han detectado.

Modo de empleo

A continuación procedemos a dar unos consejos básicos tanto para la instalación como para la configuración de AJAXTags.

Instalación

Paso 1: Copiar el fichero jar AjaxTags

Copiar el fichero ajaxtags-1.3.jar en el directorio WEB-INF/lib.

Paso 2: Definir Referencia TLD

Nota: este paso no es necesario para Servlet 2.4 containers.

Para usuarios de JSP 1.x, copiar el fichero ajaxtags.tld en el directorio WEB-INF e incluir una referencia taglib en el fichero web.xml (el fichero ajaxtags.tld se encuentra en la carpeta META-INF de ajaxtags-1.3.jar).

```
<taglib>
                <uri>http://ajaxtags.org/tags/ajax</uri>
                <location>/WEB&#45;INF/ajaxtags.tld</location>
</taglib>
```

Paso 3: Crear el manejador del lado del servidor

Debe haber un manejador que procese la petición AJAX. ##ste puede ser un servlet, una acción struts, otro JSP, o incluso un componente no Java. El único requisito es que este componente devuelva un fichero XML que muestre una lista de resultados basada en una solicitud generada por un cliente AJAX.

Aquí se puede ver un ejemplo de una clase.

```
<?xml version=

"1.0" encoding=

"UTF&#45;8"?>
        <ajax&#45;response>
                <response>
                        <item>
                                <name>Record 1</name>
                        <value&#621;</value>
                        </item>
                        <item>
                                <name>Record 2</name>
                                 <value&#622;</value>
                        </item>
                        <item>
                                <name>Record 3</name>
                                <value&#623;</value>
                        </item>
                </response>
        </ajax&#45;response>
```

Paso 4: Declarar Referencia TLD en JSP

Para cada JSP en el que se quiera usar AJAX tag, hay que definirlo al igual que con el resto de librerías tag.

```
<%@ taglib uri=

"http://ajaxtags.org/tags/ajax" prefix=

"ajax" %>
```

Paso 5: Incluir JavaScript Core

Cuando usemos AJAX, es inevitable el uso de Javascript. Se necesitan algunos ficheros de javascript:

- La librería de ajaxtags: ajaxtags.js
- La librería prototype.js
- La librería scriptaculous.js
- Las librerías overlibs.

Algo importante a tener en cuenta es que el fichero prototype.js debe ser incluido antes que el fichero ajaxtags.js

Un ejemplo de las librerías javascript necesarias para utilizar Ajaxtags es el siguiente:

```
<script type=

"text/javascript" src=

"<%=request.getContextPath()%>/js/prototype.js"/>
<script type=

"text/javascript" src=

"<%=request.getContextPath()%>/js/scriptaculous/scriptaculous.js"/>
<script type=
```

```
"text/javascript"  src=

"<%=request.getContextPath()%>/js/overlibmws/overlibmws.js"/>
<script type=

"text/javascript"  src=

"<%=request.getContextPath()%>/js/overlibmws/overlibmws&#95;crossframe.js"/>
<script type=

"text/javascript"  src=

"<%=request.getContextPath()%>/js/overlibmws/overlibmws&#95;iframe.js"/>
<script type=

"text/javascript"  src=

"<%=request.getContextPath()%>/js/overlibmws/overlibmws&#95;hide.js"/>
<script type=

"text/javascript"  src=

"<%=request.getContextPath()%>/js/overlibmws/overlibmws&#95;shadow.js"/>
<script type=

"text/javascript"  src=

"<%=request.getContextPath()%>/js/ajax/ajaxtags.js"/>
<script type=

"text/javascript"  src=

"<%=request.getContextPath()%>/js/ajax/ajaxtags&#95;controls.js"/>
<script type=

"text/javascript"  src=

"<%=request.getContextPath()%>/js/ajax/ajaxtags&#95;parser.js"/>
```

Paso 6: Definir Estilos CSS

Muchos tags dependen fuertemente de CSS para ayudar a desarrollar su función. Por ejemplo, en el caso de la función ajax:autocomplete, CSS transforma una lista desordenada (i.e., UL > LI tags) en una lista ordenada.

Enlaces de interés

Página oficial de AJAXTags:

- http://ajaxtags.sourceforge.org/

Página de artículos y ayuda sobre AJAX

- http://www.ajaxmatters.com/

Página con consejos sobre buen uso de AJAX

- http://ajaxpatterns.org/

Pequeño tutorial aportado por Java Center

- AjaxTags.pdf

Ejemplos de uso

A continuación procedemos a mostrar un ejemplo del uso de la función ajax:autocomplete, para ver con un poco más de detalle el uso de estas librerías.

Como se ha comentado anteriormente, esta función nos va mostrando una lista de valores que coinciden con lo introducido en un formulario de texto. Esto se hace en tiempo real, por lo que a cada pulsación veremos como se va refrescando el resultado de dicha lista coincidente.

He aquí una página de ejemplo que implementa esta función:

```
<html xmlns=

"http://www.w3.org/1999/xhtml">
<head>
  <meta http-equiv=

"Content-Type" content=

"text/html; charset=ISO-8859-1" />
  <title>Ejemplo AjaxTags</title>
  <script type=

"text/javascript" src=

"<%=request.getContextPath()%>/js/prototype.js"></script>

  <script type=

"text/javascript" src=

"<%=request.getContextPath()%>/js/scriptaculous/scriptaculous.js"></script>

  <script type=

"text/javascript" src=

"<%=request.getContextPath()%>/js/overlibmws/overlibmws.js"></script>
  <script type=

"text/javascript" src=

"<%=request.getContextPath()%>/js/overlibmws/overlibmws_crossframe.js"></script>
  <script type=

"text/javascript" src=

"<%=request.getContextPath()%>/js/overlibmws/overlibmws_iframe.js"></script>
  <script type=

"text/javascript" src=

"<%=request.getContextPath()%>/js/overlibmws/overlibmws_hide.js"></script>
  <script type=

"text/javascript" src=

"<%=request.getContextPath()%>/js/overlibmws/overlibmws_shadow.js"></script>

  <script type=

"text/javascript" src=
```

```
"<%=request.getContextPath()%>/js/ajax/ajaxtags.js"></script>
  <script type=

"text/javascript" src=

"<%=request.getContextPath()%>/js/ajax/ajaxtags_controls.js"></script>
  <script type=

"text/javascript" src=

"<%=request.getContextPath()%>/js/ajax/ajaxtags_parser.js"></script>

  <link rel=

"stylesheet" type=

"text/css" href=

"<%=request.getContextPath()%>/css/ajaxtags.css" />
  <link rel=

"stylesheet" type=

"text/css" href=

"<%=request.getContextPath()%>/css/site.css" />
  <link rel=

"stylesheet" type=

"text/css" href=

"<%=request.getContextPath()%>/css/displaytag.css" />
</head>
<body>
<form>
<fieldset>
    <legend>Selección de Municipio</legend>
    <!-- ORIGEN -->
    <label

for=

"sProvincia">Provincia:</label>
    <select id=

"sProvincia">
      <option value=

"-1">Seleccione una provincia</option>
      <option value=

"1">Huelva</option>
      <option value=

"2">Sevilla</option>
      <option value=

"3">Cadiz</option>
      <option value=

"4">Cordoba</option>
      <option value=

"5">Jaen</option>
      <option value=

"6">Malaga</option>
      <option value=

"7">Granada</option>
      <option value=
```

```
"8">Almeria</option>
    </select>
    <!-- LOGO DE CARGA -->
    <span id=

"logoCarga" style=

"display:none;"><img alt=

"Indicador" src=

"<%=request.getContextPath()%>/img/indicador.gif" /> Cargando...</span>
    <!-- DESTINO -->
    <label

for=

"sMunicipio">Municipio:</label>
    <select id=

"sMunicipio" disabled=

"disabled">
      <option value=

"">Seleccione un municipio</option>
    </select>
  </fieldset>
</form>
<ajax:select
  baseUrl=

"${contextPath}/dropdown.view"
  source=

"sProvincia"
  target=

"sMunicipio"
  parameters=

"sProvincia={sProvincia}"
  preFunction=

"initProgress"
  emptyOptionName=

"Seleccione un municipio"
  postFunction=

"resetProgress"
  errorFunction=

"reportError" />
</body>
</html>
```